**tr Commands:**

1. To delete a char: $cat filename.txt | tr -d ‘a’

This option deletes characters in the set specified.

2. To remove digits from a string: $cat filename.txt | tr -d [:digit:]

This command deletes the numbers from the given string.

$cat filename.txt | tr -cd [:digit:]

This command removes all characters except digits.

3. To find the hexa-decimal(a-z, A-Z 0-9): $cat filename.txt | tr -d [:xdigit:]

This command eliminates the hexadecimal values.

$cat filename.txt | tr -cd [:xdigit:]

This command removes all characters except hexa-decimal digits.

4. To remove character from a string: $cat filename.txt | tr -d [:alpha:]

This command deletes all the characters from the given string.

$cat filename.txt | tr -cd [:alpha:]

This command removes all digits and parenthesis except characters.

5. To replace character with some other: $ echo “this is a string ” | tr ‘ia’ ‘14’

o/p-->(th1s 1s 4 str1ng)

6. To discard unwanted spaces: $ echo “this is a string” | tr -s ‘ ’ ‘ ’

o/p--> this is a string

7. To change new line character with space: $tr -s ‘\n’ ‘ ’ < file.txt

This command will change new line character with space.

8. To change {} to (): $tr ‘{}’ ‘()’ <inputfile> outputfile

This command will change {} to ()

9. To convert lower case to upper case: $cat filename.txt | tr “[a-z]” “[A-Z]”  
 $cat filename.txt | tr “[:lower :]” “[:upper :]”

This command converts lower case to upper case.

10. To Translate white-spaces to tabs: $ cat filename.txt | tr [:space:] '\t'

This Command translate white-spaces in a string to tabs.

11. To Translate braces to parenthesis: $cat filename.txt | tr “{ }” “( )”  
 $cat filename.txt | tr “hello” “hiiii” (change the characters from hello to hiiii i.e.,h-h,e-i,l-i,l-i,o-i)

These two commands helps in translating the characters and the special symbols.

**To eliminate repeated digits or characters**

**\* awk ‘!a[$0]++’**

[$0] – represents current line

It will be stored in an array ‘a’

‘++’ - increment

\* **cat file.txt | awk ‘!a[$0]++’**

Example:  **cat > file.txt cat>num.txt**

ram 123

ravan 234

laxman 123

ram 456

arun 123

ajay

ram

**cat file.txt | awk ‘!a[$0]++’ cat num.txt | awk ‘!a[$0]++’**

ram 123

ravan 234

laxman 456

arun

ajay

**Find Command:**

Different operations using find command:

1) Find and delete a file with confirmation

$ find ./ -name file\_name -exec rm -i {} \;

When this command is entered, a prompt will come for confirmation, if you want to delete the file or not. If you enter ‘Y/y’ it will delete the file. Instead, if you enter ‘N/n’ it will not delete the file.

Ex: $find /home/user/Public -name new.txt -exec rm -i {} \;

rm: remove regular file ‘/home/user/Public/new.txt’ (If you enter y or Y, it will delete the file)

2) Find and delete a file without confirmation

$ find / -name file\_name -exec rm {} \;

When this command is entered, the file will be deleted directly without asking for any confirmation.

Ex: $find /home/user/Public -name new.txt -exec rm {} \;

Note: The file gets deleted

3) Search for empty files and directories

$ find / -empty

This command find all empty folders and files in the entered directory or sub-directories.

Ex: $find /home/user/Public -empty

/home/user/Public/new.txt

/home/user/Public/new2.txt

/home/user/Public/abc

4) Find all Empty Files

$ find / -type f -empty

This command is used to find all empty files under a certain path.

Ex: $find /home/user/Public -type f -empty

/home/user/Public/new.txt

/home/user/Public/new2.txt

5) Find all Empty Directories

$ find / -type d -empty

This command is used to find all empty directories under a certain path.

Ex: $find /home/user/Public -type d -empty

/home/user/Public/abc

6) Search text within multiple files

$ find / -type f -name "\*.txt" -exec grep 'any\_word' {} \;

This command print lines which have the word specified in them and ‘-type f’ specifies the input type is a file.

Ex: $ find /home/user/Public -type f -name "\*.txt" -exec grep 'cat' {} \;

I’m a cat.

cat

1. Tree command – This command helps you list all files and directories under a specified directory.

**$ tree -a directory\_name**

1. Find a specified file – This command shows the path where the specified file is present

**$ find directory\_name -name file\_name.txt**

1. Find a list of files **-** This command shows the path where a list of files is present

**$ find directory\_name -name “\*.txt”**

Finds all the text files under a particular directory

**$ find directory\_name -name “\*t.txt”**

Finds all the text files ending with “t” under a particular directory

**$ find directory\_name -name “t\*.txt”**

Finds all the text files starting with “t” under a particular directory

1. Find list of files having specified permissions – Displays all the files under the specified directory having the specified permission

**$ find directory\_name -perm 664**